

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

---

## **BEST AVAILABLE IMAGES**

---

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

---

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) Publication number : 0 485 227 A2

(12)

## EUROPEAN PATENT APPLICATION

(21) Application number : 91310338.8

(51) Int. Cl.<sup>5</sup> : H04L 29/06

(22) Date of filing : 08.11.91

(30) Priority : 08.11.90 JP 304850/90

(43) Date of publication of application :  
13.05.92 Bulletin 92/20

(64) Designated Contracting States :  
DE FR GB

(71) Applicant : NINTENDO CO., LTD.  
60, Fukuinekamitakamatsu-cho,  
Higashiyama-ku  
Kyoto-shi, Kyoto-fu (JP)

(72) Inventor : Okada, Satoru  
13-47, Fukakusananmei-cho, Fushimi-ku  
Kyoto-shi, Kyoto-fu (JP)  
Inventor : Yamagami, Hitoshi  
213, Tsuji-cho  
Ikoma-shi, Nara-ken (JP)  
Inventor : Yamano, Katsuya  
8-309, Shinomiya, Koganezuka, Yamashina-ku  
Kyoto-shi, Kyoto-fu (JP)

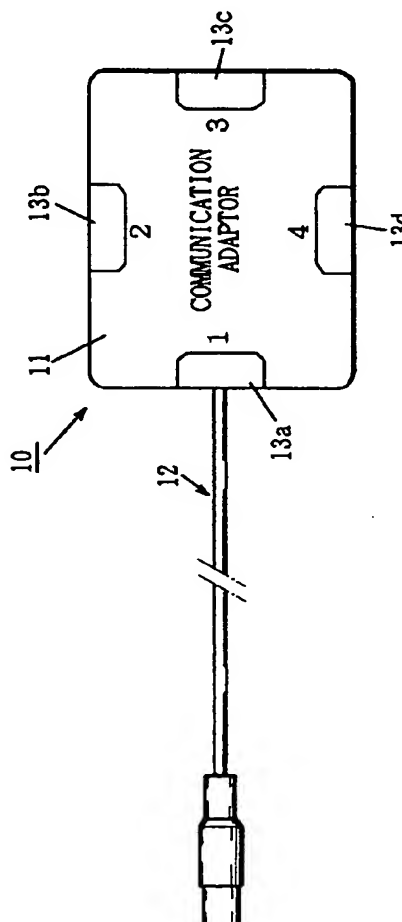
(74) Representative : Smith, Martin Stanley et al  
Stevens, Hewlett & Perkins 1 St. Augustine's  
Place  
Bristol BS1 4UD (GB)

(54) Communication adaptor for game set.

(57) The same clock signal is applied to respective bits in a clock output port in synchronization with the transmission/reception timing. A buffer memory for temporarily storing transmit/receive data stores transmit data and receive data in synchronization with the clock signal. Data converting means comprising a CPU and a ROM format-converts bit serial data received from each of game sets into bit parallel data or format-converts bit parallel data to be transmitted into bit serial data. Receive data writing means writes the data format-converted by the format converting means into transmit/receive data temporarily storing means in synchronization with the clock signal. Transmit data supplying means supplies the transmit data stored in the transmit/receive data temporarily storing means to each of bits in a data output port after being format-converted into bit corresponding data by the format converting means in synchronization with the clock signal.

Since data can be reliably transmitted in a simple circuit without using high-cost modems and the number of game sets to be connected can be increased, therefore, the present invention is applicable to a variety of games.21

FIG. 1



## BACKGROUND OF THE INVENTION

### Field of the Invention

The present invention relates generally to a communication adaptor for a game set, and more particularly, to a communication adaptor for a game set used for transmitting and receiving data related to a game in the case of, for example, doing battle (simultaneous play) with a plurality of game sets being connected thereto.

### Description of the Prior Art

In the case of data communication using personal computers, communication has been conventionally established through a long-distance transmission line and exchange. Accordingly, a transmission error is liable to occur. Consequently, high-cost modems for transmitting and receiving data and controlling the error in accordance with the transmission control procedure have been used.

On the other hand, the applicant of the present application has proposed as a technique for playing a battle game between two game sets by transmitting and receiving data related to a game in a simple method one for playing a battle game with two portable dot matrix liquid crystal game sets (trade name "GAME BOY") being connected to each other.

In the technique for establishing data communication using modems, data can be transmitted accurately but at high cost. Accordingly, it is not suitable for transmission of data on a battle game in a game set requesting cost reduction. In addition, if an attempt is made to reduce the transmission error using error control or the like, it takes long to perform processing therefor. Consequently, time required to perform game processing is restricted, to spoil the fun of the battle game. Furthermore, one modem is connected to each of the personal computers, and the modem and the personal computer establish one-to-one communication. Accordingly, it is impossible for a plurality of persons not less than three to play the battle game.

On the other hand, in the above described technique for playing a battle game with two portable dot matrix liquid crystal game sets being connected to each other, data can be transmitted at low cost. When during the game in one of the game sets, data is sent from the other game set and then, the next data is sent before the next vertical blanking period, however, the previous data disappears, so that data transmission may not, in some cases, be accurately made. The reason for this is that the two game sets differ in the progress of the game as well as the timing when vertical blanking occurs. Furthermore, in this technique, the battle game can be only enjoyed between two game sets. Accordingly, such a game as to progress while being played by not less than three persons (for

example, mah-jong or a card) cannot be achieved.

## SUMMARY OF THE INVENTION

Therefore, a primary object of the present invention is to provide a new low-cost communication adaptor for a game set of simple construction which is so adapted that a plurality of players can enjoy a battle game while being related to one another.

Briefly stated, the present invention provides a communication adaptor for a game set (which corresponds to 10 in the embodiment), which is used with a plurality of game sets for playing a game while being related to the game state of the other game sets being thereto, for transmitting and receiving data related to the game between a plurality of game sets. Each of the game sets 20a to 20d is provided with a first serial interface 26 for inputting and outputting data on the game and a buffer memory 263 for temporarily storing transmit/receive data, and the game sets are relevantly connected to one another.

The communication adaptor 10 for a game set comprises a housing 11, a plurality of second connecting tools 13a to 13d, a data input port 181, a data output port 182, a clock output port 183, clock signal generating means 14 and 17, format converting means 14 and 15, receive data writing means 14 and 15, and transmit data supplying means 14 and 15.

Each of the second connecting tools is provided for the housing and includes at least a first terminal, a second terminal and a third terminal, the second connecting tool being connected to the first serial interface in each of the game sets. The data input port has storing portions of bits whose number corresponds to the number of a plurality of connectors so as to store serial data inputted from the connectors, the storing portions corresponding to the bits being respectively connected to the first terminals included in the second connecting tools. The data output port has storing portions of bits whose number corresponds to the number of the plurality of second connecting tools, the storing portions corresponding to the bits being respectively connected to the second terminals included in the second connecting tools. The clock output port has storing portions of bits whose number corresponds to the number of a plurality of connectors such that it can generate clocks required for the connectors to input and output a signal, the storing portions corresponding to the bits being respectively connected to the third terminals included in the second connecting tools.

The clock signal generating means applies the same clock signal used for data communication with the game sets to each of the bits in the clock output port in synchronization with the transmission/reception timing. The transmit data temporarily storing and receive data temporarily storing means stores transmit data on the game to be transmitted to each of the

game sets and receive data on the game received from each of the game sets in synchronization with the clock signal. The format covering means format-converts the serially arranged receive data received from each of the game sets into bit parallel data or format-converts the bit parallel data to be transmitted into serial data. The receive data writing means writes into the transmit/receive data temporarily storing means bit corresponding data out of data in the data input port which are received from each of the game sets after being format-converted by the format converting means in synchronization with the clock signal. The transmit data supplying means reads out the transmit data stored in the transmit/receive data temporarily storing means to supply the same to each of the bits in the data output port after being format-converted into bit corresponding data by the format converting means so as to transmit the transmit data stored in the transmit/receive data temporarily storing means to each of the game sets in synchronization with the clock signal. That is, the transmit data supplying means outputs the transmit data by performing the work reverse to that performed by the receive data writing means

According to the present invention, data related to a battle game can be reliably transmitted between a communication adaptor and a plurality of game sets in a simple circuit without using high-cost modems, and the number of game sets which can be connected to the communication adaptor can be increased. Accordingly, it is possible to diversify the type of the battle game to which the present invention is applied.

The foregoing and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing the appearance of a communication adaptor for a game set according to one embodiment of the present invention;

Fig. 2 is an illustration of a state where the communication adaptor is used;

Fig. 3 is a block diagram showing one embodiment of the communication adaptor;

Fig. 4 is a diagram showing a map of a RAM (transmit/receive data temporarily storing means) included in the communication adaptor;

Fig. 5 is a block diagram showing one example of a game set connected to the communication adaptor;

Figs. 6A and 6B are diagrams illustrating a state where data communication is established between the communication adaptor and the game sets, where Fig. 6A shows operation modes of the communication adaptor, and Fig. 6B shows the

relationship between each of RAM areas storing transmit/receive data and the flow of data in the state where data communication is established; Figs. 7A to 7F are flow charts for explaining operations of the communication adaptor, where Fig. 7A shows a case where power is turned on (at the start time), Fig. 7B shows a case at the time of a timer interruption, Fig. 7C shows a case at the time of a connection information mode, Fig. 7D shows a case at the time of a start mode, Fig. 7E shows a case at the time of a data communication mode, and Fig. 7F shows a case at the time of a reset mode;

Figs. 8A to 8B are flow charts related to a communicating operation on the side of the game set, where Fig. 8A shows a case of a main program, and Fig. 8B shows a case at the time of an SIO interruption; and

Figs. 9A to 12B are illustrations of a data format for each operation mode, where Figs. 9A to 9C show a case in a connection information mode, Figs. 10A and 10B show a case in a start mode, Figs. 11A to 11B show a case in a data communication mode, and Figs. 12A and 12B show a case in a reset mode.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 is a diagram showing the appearance of a communication adaptor for a game set (hereinafter referred to as "communication adaptor") according to one embodiment of the present invention. In Fig. 1, a communication adaptor 10 comprises a housing 11 in the shape of a polygon having not less than three sides (for example, a rectangle or a square in the drawings). A code with a plug 12 for connecting the communication adaptor 10 to a certain game set 20a serving as a child set (which may, in some cases, be illustrated as "DMG" because the game set uses a dot matrix liquid crystal display) is pulled out of the side surface of the housing 11 through a connecting terminal 13a.

Connectors 13b to 13d for connecting the communication adaptor 10 to a plurality of other game sets 20b to 20d by a code 12' are mounted on the other side surfaces of the housing 11. Each of the connecting terminal 13a and the connectors 13b to 13d includes, for example, first and second terminals for serially transmitting transmit/receive data on one bit basis and a third terminal for transmitting a clock signal, and includes two terminals, that is, a plus terminal of the power supply and a minus terminal thereof such that it is supplied with power from any one of the game sets as required.

When a plurality of players enjoy a battle game, connection is made as shown in Fig. 2, and data required for the battle game in the game sets 20a to

20d is transmitted to the game sets through the communication adaptor 10. Consequently, the communication adaptor 10 has a data transmitting and receiving function.

Although in Fig. 2, four game sets (child sets) are connected to the communication adaptor 10 assuming a case where a maximum of four persons simultaneously play the battle game, the connector 13d is not required if the communication adaptor is for a maximum of three persons. In addition, if five or more persons play the battle game, connectors whose number corresponds to the number of game sets to be connected are added. In this case, used as the housing 11 may be one in a pentagonal shape or one in an oblong shape having one side provided with a plurality of connectors. Furthermore, the code with a plug 12 for the first game set 20a may be connected for the other game sets 20b to 20d in place of the connectors. Additionally, since in this example, the communication adaptor 10 has no power by itself, a code with a plug through the connecting terminal 13a for supplying power to the communication adaptor 10 is connected to one game set 20a to be a center. If power is supplied to the communication adaptor 10 by itself, however, a connector may be provided in place of the connecting terminal 13a and the communication adaptor 10 may be connected to the game set 20a using a code 12'.

Fig. 3 is a block diagram showing one embodiment of the communication adaptor. The communication adaptor 10 comprises a microprocessor (hereinafter referred to as "CPU") 14. A read-only memory (hereinafter referred to as "ROM") 15 and a random access memory (hereinafter referred to as "RAM") 16 are connected to the CPU 14 through a data bus and an address bus. The ROM 15 stores program data for achieving operations of flow charts shown in Figs. 7A to 7F as described later.

The RAM 16 is used as an input/output buffer or a transmission/reception buffer for temporarily storing input/output data. More specifically, the RAM 16 comprises a transmission buffer region 161, a reception buffer region 162, a register region 163, and a flag region 164, as shown in Fig. 4. The transmission buffer region 161 stores respective 1-byte data on the game in the game sets 20a to 20d by four bytes (four bytes to be transmitted during a period of eight clocks from terminals corresponding to respective bits in an output port 182) for each game set and has a storage capacity of at least  $4 \times 4 = 16$  bytes. Similarly, the reception buffer region 162 stores respective 1-byte data on the game in the game sets 20a to 20d by four bytes (four bytes received during a period of eight clocks to terminals corresponding to respective bits in an input port 181) for each game set and has a storage capacity of at least  $4 \times 4 = 16$  bytes. The register region 163 comprises, for example, a register 163s for temporarily storing communication cycle data (S), a

register 163t for temporarily storing data (T) for controlling one byte and one-byte intervals, and a register 163n for temporarily storing the number of data (n). The flag region 164 comprises a timer interruption flag, a mode flag and the like.

Furthermore, a clock oscillator 17, the input port 181, the output port 182, and a clock output port 183 are connected to the CPU 14. The clock oscillator 17 generates a reference clock signal having a frequency correlated with a machine cycle of the CPU 14 to apply the same to the CPU 14. The CPU 14 operates in a cycle based on the reference clock signal, and generates one pulse of a clock signal for synchronization of the transmission/reception timing to apply the same to the connecting terminal 13a and the connectors 13b to 13d corresponding to the game sets 20a to 20d every time it outputs 1-bit transmit data or receives receive data. Consequently, the clock oscillator 17 and the CPU 14 constitute clock signal generating means.

The input port 181 has storing portions of four bits respectively corresponding to the game sets 20a to 20d (for example, D-type flip-flops), and temporarily stores in the storing portions data transmitted serially by bit from the game sets 20a to 20d through the respective first terminals (not shown) of the connecting terminal 13a and the connectors 13b to 13d. Similarly, the output port 182 has storing portions of four bits, temporarily stores in the storing portions data to be transmitted serially by bit to the game sets 20a to 20d, and transmits the data to the game sets 20a to 20d serially by bit through the respective second terminals (not shown) of the connecting terminal 13a and the connectors 13b to 13d. The clock output port 183 has storing portions of four bits, and alternately stores in the storing portions logic "1" and logic "0" in synchronization with the transmission/reception cycle of 1-bit data to supply a clock signal synchronized with the transmission/reception timing to the game sets 20a to 20d through the respective third terminals (not shown) of the connecting terminal 13a and the connectors 13b to 13d.

Furthermore, a timer 19 constituted by a hardware circuit (hereinafter referred to as "hardware timer") 19 capable of presetting its set time for determining a communication cycle. The CPU 14 also has a software timer function so as to know one-byte intervals (time T) in a software manner by program processing.

Fig. 5 is a block diagram showing each of the game sets connected to the communication adaptor. Each of the game sets 20a to 20d comprises a liquid crystal display (LCD) 21, a display driving circuit 22, a display control circuit 23, an operation portion 24, a connector 25, and a communication control circuit 26. An external memory cartridge 30 storing various program data for game control, display control and communication control is detachably connected to the

connector 25 in each of the game sets 20a to 20d.

The liquid crystal display 21 is of, for example, a dot matrix type and is driven by a segment driver 221 and a common driver 222 included in the display driving circuit 22. The display control circuit 23 comprises a CPU core 231, an input/output port 232, a direct memory access (DMA) controller 233, a line buffer 234, an LCD controller 235, an LCD drive signal buffer 236, a display RAM 237, and an RAM interface 238. The details of the circuits are described in Japanese Patent Publication No. 4453/1989 (Japanese Patent Laid-Open Gazette No. 210562/1990) related to the application of the present applicant and hence, are omitted.

The communication control circuit 26 comprises a shift register 261, an interrupt condition detecting circuit 262, and a transmission and reception buffer RAM 263. The shift register 261 performs a shifting operation by a clock signal (SCK) from the communication adaptor 10, and serves to load 1-byte transmit/receive data in parallel and convert the transmit/receive data into bit serial data to output the same under the control of the CPU core 231 or to convert the inputted serial data into parallel data while shifting the serial data to apply the same to the CPU core 231. The interrupt condition detecting circuit 262 generates an interrupt signal every time eight clock signals (SCK) are applied (that is, 1-byte data is transmitted and received), to apply the same to the CPU core 231. The transmission/reception buffer RAM 263 comprises a reception buffer region and a transmission buffer region. The reception buffer region and the transmission buffer region respectively comprise bytes whose number corresponds to the number of the game sets connected to the communication adaptor 10 (for example, four bytes). In addition, a connector 27 is provided in relation to the communication control circuit 26, and terminals of the connector 27 are respectively connected to the most significant bit, the least significant bit, and a clock signal (SCK) line of the shift register 261.

Figs. 6A and 6B are diagrams illustrating the principle of a state where data communication is established between the communication adaptor and the game sets, where Fig. 6A shows operation modes of the communication adaptor, and Fig. 6B shows the relationship between each of RAM areas for storing transmit/receive data and the flow of data in the state where data communication is established.

Each of the game sets only receives data on one game sent by making transmission once. Accordingly, communication must be repeated four times so as to exchange data of one byte on each of the four game sets between the game sets. Each of the game sets transmits data of one byte, while receiving data of four bytes. Transmission and reception are simultaneously made. Accordingly, each of the respective game sets transmits necessary data of one byte and

dummy data (in which eight bits are all zero) of three bytes, to obtain data on the four game sets.

In Fig. 6B, parts corresponding to the first byte, the second byte, the third byte, and the fourth byte in the flow of the transmit/receive data are respectively indicated by a thick line, a thin line, a dotted line, and a one-dot and dash line.

The operation modes will be schematically described with reference to Fig. 6A. When power is turned on, that is, the code with a plug 12 is connected to the connector 27 in the game set 20a, the CPU 14 in the communication adaptor 10 executes a connection information mode (which will be described in detail with reference to Fig. 7C and Figs. 9A to 9C as described later) so as to confirm the connecting state. When any one of the game sets 20a to 20d is connected to the communication adaptor 10 and the game is started, the CPU 14 executes a start mode (which will be described in detail with reference to Fig. 7D and Figs. 10A and 10B as described later). During a period of the game, the CPU 14 executes a data communication mode (which will be described in detail with reference to Fig. 7E and Figs. 11A and 11B as described later). In addition, when it is desired to reset the game set during the game, for example, when it is desired to play the game again from the beginning without turning off power, the CPU 14 executes a reset mode (which will be described in detail with reference to Fig. 7F and Figs. 12A and 12B as described later).

Referring to Fig. 6B, transmit data representing the game state required for the battle game is stored in the first byte in an area corresponding to each of the game sets 20a to 20d in the RAM 263 (transmission buffer region) 263.

Transmit data of one byte for each of the game sets 20a to 20d is bit serial data arranged in the order from B0 to B7, and corresponding bit data for the game sets 20a to 20d are respectively transmitted through the first terminals of the connecting terminal 13a and the connectors 13b to 13d in parallel by four bits. This operation is performed eight times, thereby to transmit 1-byte data. This data is read by the CPU 14, and is format-converted into data of one byte comprising eight bits on the basis of programs of the CPU 14 and the ROM 15, to be written into a corresponding area in the reception buffer region 161.

This operation is repeated until transmit data of four bytes are transmitted.

However, data in the second to fourth bytes are dummy data in which eight bits are all zero, for example, when data in communication established once is only one byte. The foregoing is one cycle of communication.

Such a series of operations will be described in more detail. First, transmit data of only one byte are respectively sent from the game sets 20a to 20d to the first terminals of the connecting terminal 13a and the

connectors 13b to 13d. Each of the transmit data is transmitted one bit at a time from its least significant bit. Since four game sets are connected to a communication adaptor in the present embodiment, transmit data of a total of four bits are inputted. The transmit data of four bits are inputted once through the first terminals of the terminal 13a and the connectors 13b to 13d in storing portions directly connected to the connecting terminal and the connectors. That is, only the respective least significant bits of data on the four game sets are inputted as a collection of four bits in the storing portions. Thereafter, the respective one bits are format-converted into parallel data, and are further stored again in the receive data temporarily storing means. When this work is terminated, the respective second least significant bits are transmitted from the game sets in the same manner as the previous process. The bits are inputted as a collection of four bits, and the respective one bits are format-converted into parallel data and then, are stored again in the receive data temporarily storing means in the same manner as the previous process. This process is repeated eight times, to constitute one cycle of communication. As a result, 8-bit (1-byte) data is transmitted from each of the game sets, and 1-byte data sent from each of the game sets is stored as parallel data in the receive data temporarily storing means.

On the other hand, when the CPU 14 receives data received in reception made once, the receive data in the reception buffer region 162 are transmitted or written into respective addresses in areas for the game sets 20a to 20d in the transmission buffer region 161. At this time, data representing the state of the battle game of the four game sets 20a to 20d is required in each of the game sets. Accordingly, in order to transmit the received data on the four game sets to the four game sets by the transmission of the 4-byte data, the CPU 14 respectively writes data for 20a, data for 20b, data for 20c, and data for 20d into the first byte, the second byte, the third byte and the fourth byte in areas corresponding to the game sets 20a to 20d in the transmission buffer region 161. Since the same data must be transmitted to the four game sets, the same data are respectively stored into the areas corresponding to the game sets 20a to 20d in the transmission buffer region 161.

Thereafter, data in each of the first to fourth bytes in an area corresponding to each of the game sets in the transmission buffer region 161 is transmitted to each of the game sets 20a to 20d serially (that is, in the order of B0, B1, ..., B7, ... in the first byte for the game sets 20a to 20d, ..., and B0, B1, ..., B7, ... in the fourth byte for the game sets 20a to 20d) in the order from the first to the fourth bytes. The data thus received is stored in the reception buffer region 162 in the transmission/reception buffer RAM 263 on the side of each of the game sets 20a to 20d.

Figs. 7A to 7F are flow charts for explaining oper-

ations of the communication adaptor, where Fig. 7A shows a case where power is turned on, Fig. 7B shows a case at the time of a timer interruption, Fig. 7C shows a case at the time of a connection information mode, Fig. 7D shows a case at the time of a start mode, Fig. 7D shows a case at the time of a data communication mode, and Fig. 7F shows a case at the time of a reset mode.

Figs. 8A to 8B are flow charts for explaining the communicating operation of the game set, where Fig. 8A shows a case of a main program, and Fig. 8B shows a case at the time of an SIO interruption.

Figs. 9A to 12B are illustrations of a data format for each mode, where Figs. 9A to 9C show a case in a connection information mode, Figs. 10A and 10B show a case in a start mode, Figs. 11A to 11B show a case in a data communication mode, and Figs. 12A and 12B show a case in a reset mode.

Description is now made of operations performed when data communication is established between the communication adaptor and the game sets for each mode.

#### (1) Power on processing and timer interruption processing

When the plug of the code 12 is inserted into the game set 20a so that a power switch (not shown) of the game set 20a is turned on, power is supplied to the communication adaptor 10. Correspondingly, the CPU 14 starts an operation performed when power is turned on as shown in Fig. 7A. That is, in the step (abbreviated as "S" in the drawings) 1, a communication cycle (S) time is set in the hardware timer 19. In the subsequent step 2, a timer interruption flag is set. In the step 3, zero is set in a mode flag. In the step 4, the CPU 14 waits until a timer interruption occurs.

#### (2) Connection information mode

If a timer interruption occurs while the CPU 14 waits in the step 4, the program proceeds to timer interruption processing shown in Fig. 7B. That is, it is judged in the step 5 whether or not the mode flag is zero. Since the mode flag is zero immediately after power is turned on, a connection information mode is executed in the step 10 (see Fig. 7C). The processing in this connection information mode is performed for the purpose of transmitting and receiving data between only the game sets requiring transmission and reception by knowing which of the connecting terminal 13a and the connectors 13b to 13d is connected to each of the game sets 20a to 20d.

More specifically, in the step 11, connection information data to be transmitted to each of the game sets 20a to 20d as shown in Fig. 9A is set in the transmission (output) buffer region 161. A combination which corresponds to three bits of data for knowing which of



the connecting terminal 13a and the connectors 20b to 20d is connected to each of the game sets 20a to 20d and data for knowing which of the connecting terminal and the connectors is connected to the game set at the present time is added behind a head end code (a code indicating the head end of data; FE in hexadecimal notation: which is marked with a sign \$ in front of data in hexadecimal notation in the drawings) (the same data is sent three times). In this 1-byte data, high order digits in hexadecimal notation, that is, high order four bits D4 to D7 correspond to the game sets 20a to 20d, and each of the game sets is connected to the communication adaptor 10 when each of the bits is "1" while not being connected to the communication adaptor 10 when it is "0". Low order digits in hexadecimal notation, that is, low order four bits designate the game sets 20a to 20d using binary numbers.

In the step 12, connection information data (see Fig. 9c) set to four bytes in the area corresponding to the game set 20a in the transmission buffer region 161 is first format-converted, to be applied to the connector 27 in each of the game sets through the output port 182 and the connecting terminal 13a serially by bit. At this time, the CPU 14 generates one clock signal (SCK) for synchronization of the transmission/reception timing every time it transmits 1-bit data, to apply the same to each of the storing portions in the clock output port 183. Correspondingly, the clock signal is applied to the connector 27 in the game set 20a through the connecting terminal 13a. Consequently, data in the shift register 261 in the game set 20a is shifted one bit, and the interrupt condition detecting circuit 262 is counted up by one. When this operation is performed eight times (by one byte), the interrupt condition detecting circuit 262 generates an interrupt signal in response to the detection of eight clocks, to apply the same to the CPU core 231. Correspondingly, the game set connected performs processing shown in Fig. 8B. The details thereof will be described later.

In the above described manner, when the transmission of the 4-byte connection information data to be transmitted to the game set 20a is terminated, connection information data are sequentially transmitted to the other game sets 20b to 20d in the same manner and then, the program proceeds to the next step 13.

In the step 13, each of the game sets 20a to 20d transmits to the communication adaptor 10 4-byte data comprising a connection recognition code (\$88) of two bytes, a connect time code (a code designating time T at one-byte intervals for communication and a communication cycle (time required to terminate communication once) S, a code designating the number of bytes of data to be transmitted by each of the game sets in communication established once (any one of 1 to 4 which is the same in all the game sets) so as to inform that one's own is connected in response to the

reception of the connection information data (see Fig. 9B). When the input port 181 receives the data, the CPU 14 format-converts the receive data and writes the same into the area corresponding to the game set 20a in the reception buffer region 162, thereby to store the receive data in the step 14.

It is judged in the subsequent step 15 whether or not the transition to the start mode is to be made; otherwise communication cycle data (S) and communication time data (T) are set in the register region 163 in the step 16 and then, the program is returned to the normal operation step (the step 4).

If there is a start request code (for example, \$AA; see Fig. 10A) after a timer interruption operation is repeated several times, it is judged in the above described step 15 that the transition to the start mode is to be made, so that 1 is set in the mode flag in the subsequent step 17.

### (3) Start mode

When a timer interruption occurs while the CPU 14 waits in the above described step 4, it is judged in the step 5 that the mode flag is not zero, so that the program proceeds to the step 6. It is judged in the step 6 that the mode flag is 1, so that the program proceeds to a subroutine (the step 20) in the start mode.

In the start mode, the communication cycle data S is set in the hardware timer 19 as a constant of the hardware timer in the step 21, as shown in Fig. 7D. In the subsequent step 22, the communication time data T is set as a constant of a software timer. In addition, the constant S of the hardware timer 19 and the constant T of the software timer are selected to be  $S > T \cdot n$  when n is taken as the number of times of communication (where n is any one of 4, 8, 12 and 16). The reason for this is that a communication cycle time (time elapsed until communication is terminated n times) is managed by the constant (S) of the hardware timer, and time required to establish communication once is managed by the constant (T) of the software timer. More specifically, assuming that the time (T) required to establish communication once is 1 ms, n is any one of 4, 8, 12 and 16. Accordingly, the value of S is necessarily not less than 4, 8, 12 or 16 with respect to n. Unless the value is thus set, the communication cycle is terminated before communication is terminated a required number of times, so that the game sets judge that communication is terminated before the communication is entirely terminated, resulting in asynchronization between the game sets and the communication adaptor.

In the step 25, data on the number of times of communication (n) is set in the register. In the subsequent step 26, 2 is set in the mode flag, to specify the execution of the data communication mode. In the step 27, start codes (for example, \$CC codes of four bytes) shown in Fig. 10B are transmitted. Thereafter,



the program is returned to the step 4.

#### (4) Data communication mode

If a timer interruption occurs while the CPU 14 waits in the above described step 4, it is judged in the step 5 that the mode flag is not zero, and it is judged in the step 6 that the mode flag is not 1, so that the program proceeds to the step 7. It is judged in the step 7 that the mode flag is 2, so that the program proceeds to a subroutine in the data communication mode (the step 30).

In the data communication mode, transmit data (see Fig. 11B) related to the battle game in the transmission buffer region 161 is transmitted and at the same time, a clock signal is outputted every time 1-bit data is transmitted in the step 31, as shown in Fig. 7E, in the same manner as that in the above described step 12 in the connection information mode. In this data communication, the order of transmission is changed as shown in Fig. 11B depending on which of one to four bytes the length of data to be transmitted in transmission made once is.

In the next step 32, each of the game sets 20a to 20d transmits data shown in Fig. 11A to the communication adaptor 10 in response to the reception of the data related to the game. In this case, each of the game sets adds dummy data having no meaning behind the data related to the game to adapt the number of bytes of the transmit data to a predetermined type. However, the type differs depending on which of one to four bytes the length of the data is. In response thereto, the CPU 14 in the communication adaptor 10 receives the transmit data from the game set.

In the step 33, the CPU 14 converts the receive data into a format writable into the RAM 16 and then, stores the same in the reception buffer region 162. The storing manner in the case is as shown in Fig. 6B.

It is judged in the subsequent step 34 whether or not the transition to the reset mode is to be made; otherwise the data in the reception buffer region 162 is inversely format-converted to be written into the transmission buffer region 161 in the step 35 and then, the program is returned. The data written into the transmission buffer region 161 is transmitted to each of the game sets in the next timer interruption timing in repeating the above described step 31 again.

On the other hand, when a player using any one of the game sets operates a reset switch (not shown) so that a reset request code (for example, \$FF; see Fig. 12A) is transmitted, it is judged in the above described step 34 that the reset request code is transmitted, so that the program proceeds to the step 36. In the step 36, 3 is set in the mode flag and then, the program is returned.

#### (5) Reset mode

If a timer interruption occurs again while the CPU 14 waits in the above described step 4, the program proceeds to timer interruption processing. At this time, when it is judged in the step 7 that the mode flag is not 2 after the steps 5 and 6, the program proceeds to the reset mode in the step 40.

In the reset mode, the CPU 14 transmits a reset code (\$FF) as shown in Fig. 12B to each of the game sets 20a to 20d in the step 41 (see Fig. 7F). Subsequently, zero is set in the mode flag in the step 42 and then, the program is returned. Consequently, when a reset request occurs, the program is resumed from the above described connection information mode in the timer interruption processing.

#### (6) Operation on the side of game sets

Although the operation of each of the game sets 20a to 20d was briefly described in relation to the description of the operations of the communication adaptor 10, the details of the operation related to communication will be described mainly on the side of the game set with reference to Figs. 8A and 8B.

Each of the game sets 20a to 20d generally executes a main program shown in Fig. 8A. More specifically, in the step 51, the game set reads data representing a state where a key included in the operation portion 24 is operated (key input state). In the subsequent step 52, the CPU core 231 performs game processing on the basis of program data in a ROM included in the external memory cartridge 30. At this time, the program data is for the battle game and thus, the data on the battle game must be transmitted and received to and from the other game sets through the communication adaptor 10 so as to know the game state of the other players. Therefore, the CPU core 231 previously writes its own transmit data into the transmission buffer region 161. Thereafter, it is judged in the step 53 whether or not communication in one cycle is terminated. If it is judged that the communication is not terminated, the CPU core 231 waits until the termination of the communication is detected. If a communication interruption occurs from the communication adaptor 10 in this state, the program proceeds to communication interruption processing shown in Fig. 8B.

In the communication interruption processing, it is judged in the step 61 whether or not the contents of the mode register are 1. The program proceeds to the step 62 if it is judged that the contents of the mode register are not 1, while proceeding to the step 64 if it is judged that the contents of the mode register are 1. It is judged in the step 62 whether or not the reset code (\$FF) is stored in the reception buffer region 162. If it is judged that there is no reset code, the program proceeds to the step 64. On the other hand, if it is judged

that the reset code is stored, the register storing the number of times of communication (hereinafter referred to as communication number register) is reset (the value of n is set to zero) in the step 63 and then, the program proceeds to the step 64.

In the step 64, receive data is written into the transmission buffer region 161 in the transmission/reception buffer RAM 263 or transmit data to be transmitted to the communication adaptor 10 is written into the reception buffer region 162 therein. In the subsequent step 65, 1 is added to the contents of the communication number register. Thereafter, it is judged in the step 66 whether or not the contents of the communication number register coincide with the number of times of communication (n) applied from the communication adaptor 10. If it is judged that both do not coincide with each other, data on the number of times of communication is stored in the step 67, and a code instructing waiting for reception to the CPU core 231 (580) is written into the register in the step 68 and then, the program is returned to the step 53.

On the other hand, when a communication interruption occurs while the CPU core 231 waits in the step 53 so that the communicating operation is performed a designated number of times while the operation in the steps 61 to 68 is repeated, it is judged in the step 66 that both coincide with each other, so that the program proceeds to the step 69. In the step 69, zero is set in the communication number register, thereby to raise a communication termination flag to indicate that communication in one cycle is terminated. Subsequently, when the step 68 is carried out and then, the program is returned to the step 53, it is judged that communication in one cycle is terminated, so that the program is returned to the step 51. Such an operation is repeated. That is, the steps 51 and 52 are repeated for each communication in one cycle.

Although description was made of a case where four game sets are connected to the communication adaptor, it is needless to say that data is not transmitted and received to and from the game set which is not connected to the communication adaptor because transmit data is not written into a buffer area corresponding to the game set.

Although description was made of a case where the present invention is applied to a portable game set, the present invention is not limited to the same. For example, the present invention is applicable to a home television game set and the like.

Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

## Claims

1. A communication adaptor for a game set, which is used with a plurality of game sets each provided with a first connecting tool for inputting and outputting data on a game and a buffer memory for temporarily storing transmit/receive data for playing the game while being related to the game state of the other game sets being connected thereto, for transmitting and receiving the data on the game to and from the plurality of game sets, comprising:

a housing;

a plurality of second connecting tools provided for said housing and each including at least a first terminal, a second terminal and a third terminal, the second connecting tool being connected to the first connecting tool in each of the game sets;

a data input port having storing portions of bits whose number corresponds to the number of said plurality of second connecting tools, the storing portions corresponding to the bits being respectively connected to the first terminals included in the second connecting tools;

a data output port having storing portions of bits whose number corresponds to the number of said plurality of second connecting tools, the storing portions corresponding to the bits being respectively connected to the second terminals included in the second connecting tools;

a clock output port having storing portions of bits whose number corresponds the number of said plurality of second connecting tools, the storing portions corresponding to the bits being respectively connected to the third terminals included in the second connecting tools,

clock signal generating means for applying the same clock signal used for data communication with said game sets to each of the bits in said clock output port;

transmit/receive data temporarily storing means for storing transmit data on the game to be transmitted to each of said game sets and receive data on the game received from each of the game sets in synchronization with said clock signal;

format converting means for format-converting the bit serial data received from each of said game sets into bit parallel data or format-converting the bit parallel data to be transmitted into bit serial data;

receive data writing means for writing into said transmit/receive data temporarily storing means bit corresponding data out of data in said data input port which are received from each of said game sets after being format-converted by said format converting means in synchronization with said clock signal; and

transmit data supplying means for reading out the transmit data stored in the transmit/receive data temporarily storing means to supply the same to each of the bits in said data output port after being format-converted into bit corresponding data by said format converting means so as to transmit the transmit data stored in said transmit/receive data temporarily storing means to each of said game sets in synchronization with said clock signal.

5

10

2. The communication adaptor for a game set according to claim 1, wherein

said transmit/receive data temporarily storing means comprises a transmission buffer region and a reception buffer region,

15

said reception buffer region including buffer areas respectively corresponding to said plurality of game sets, each of the buffer areas being divided into different addresses, that is, an address storing transmit data for one game set and addresses storing dummy data corresponding to the other game sets,

20

said transmission buffer region including buffer areas respectively corresponding to said plurality of game sets, each of the buffer areas being divided into addresses storing transmit data for the respective game sets.

25

3. A communication adaptor for a plurality of game sets each provided with a first interface for transferring game data to and from the game set; the communication adaptor comprising a plurality of second interfaces for connection respectively to said first interfaces, each of said second interfaces having at least a first, a second and a third terminal, the communication adaptor further having a data input port coupled to the first terminals of said second interfaces, a data output port coupled to the second terminals of said second interfaces and a clock output port coupled to the third terminals of said second interfaces, there being a clock signal generating means whereby clock signals are applied via the said clock output port to synchronise transmission and reception of game data between game sets connected to the second interfaces.

30

35

40

45

50

55

10

FIG. 1

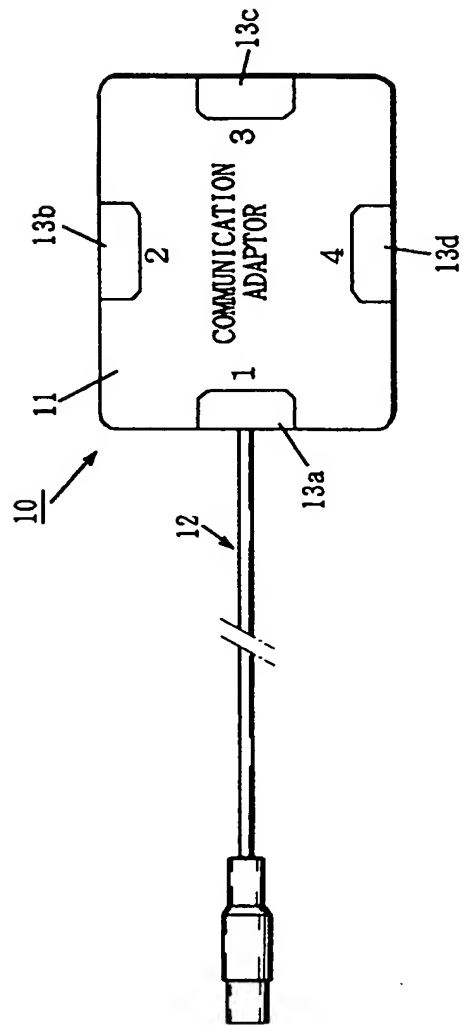


FIG. 2

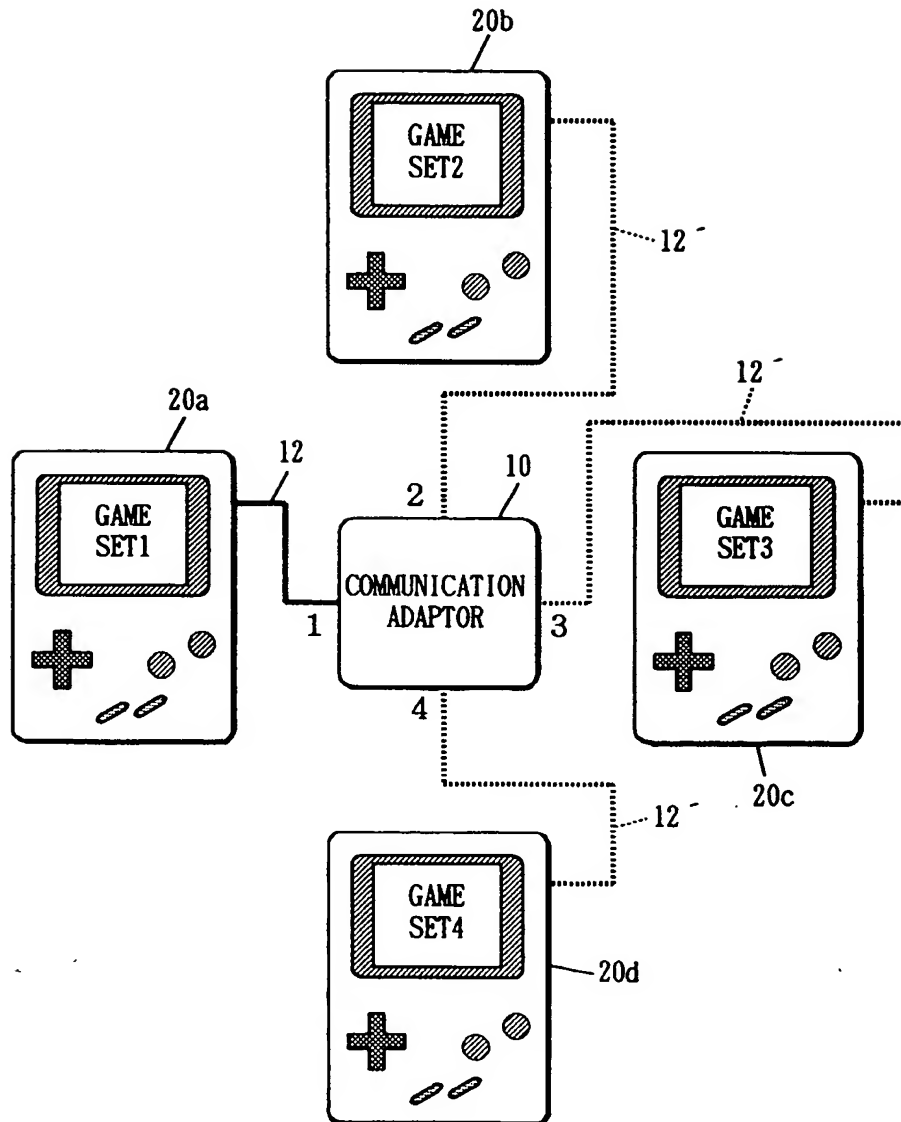


FIG. 3

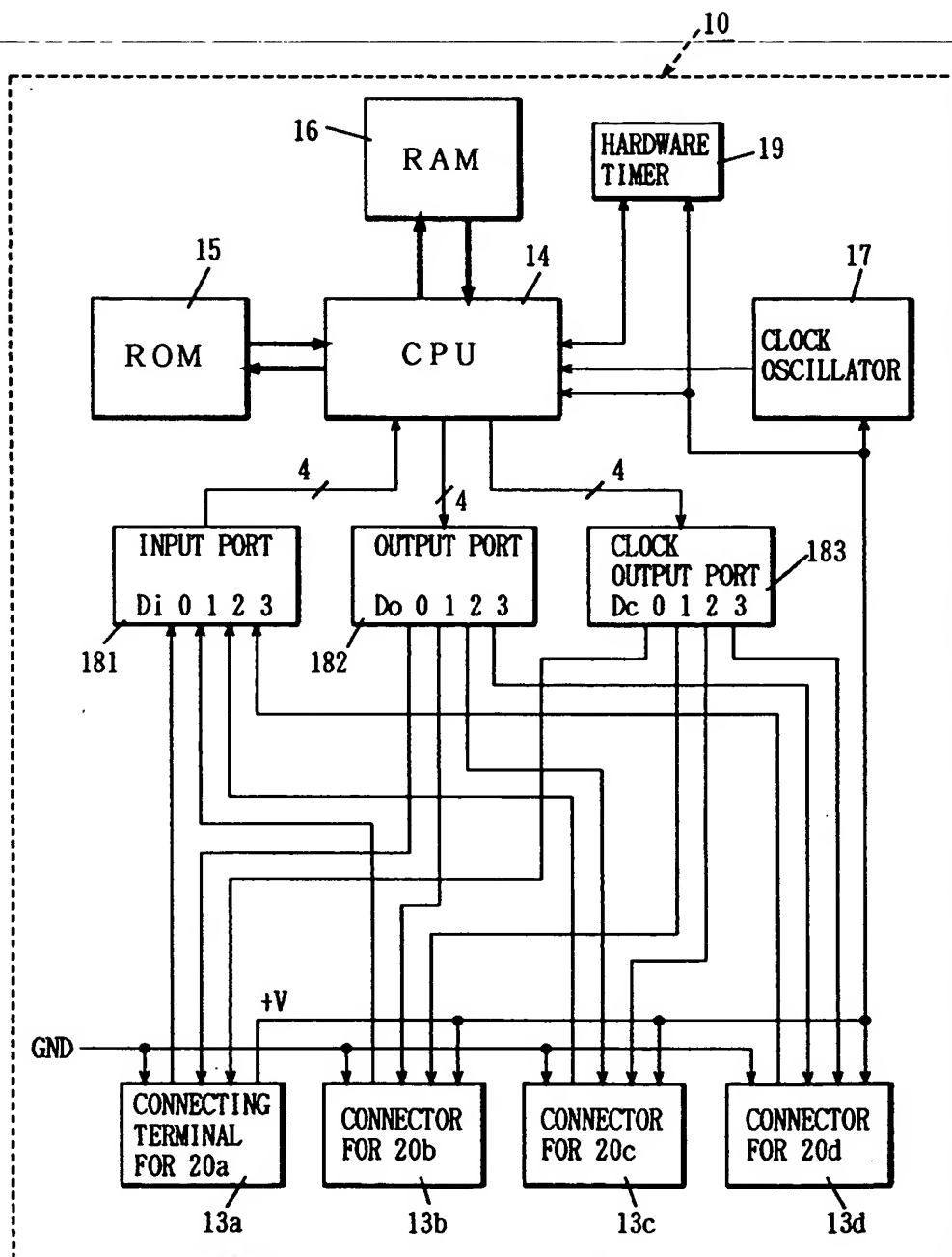


FIG. 4

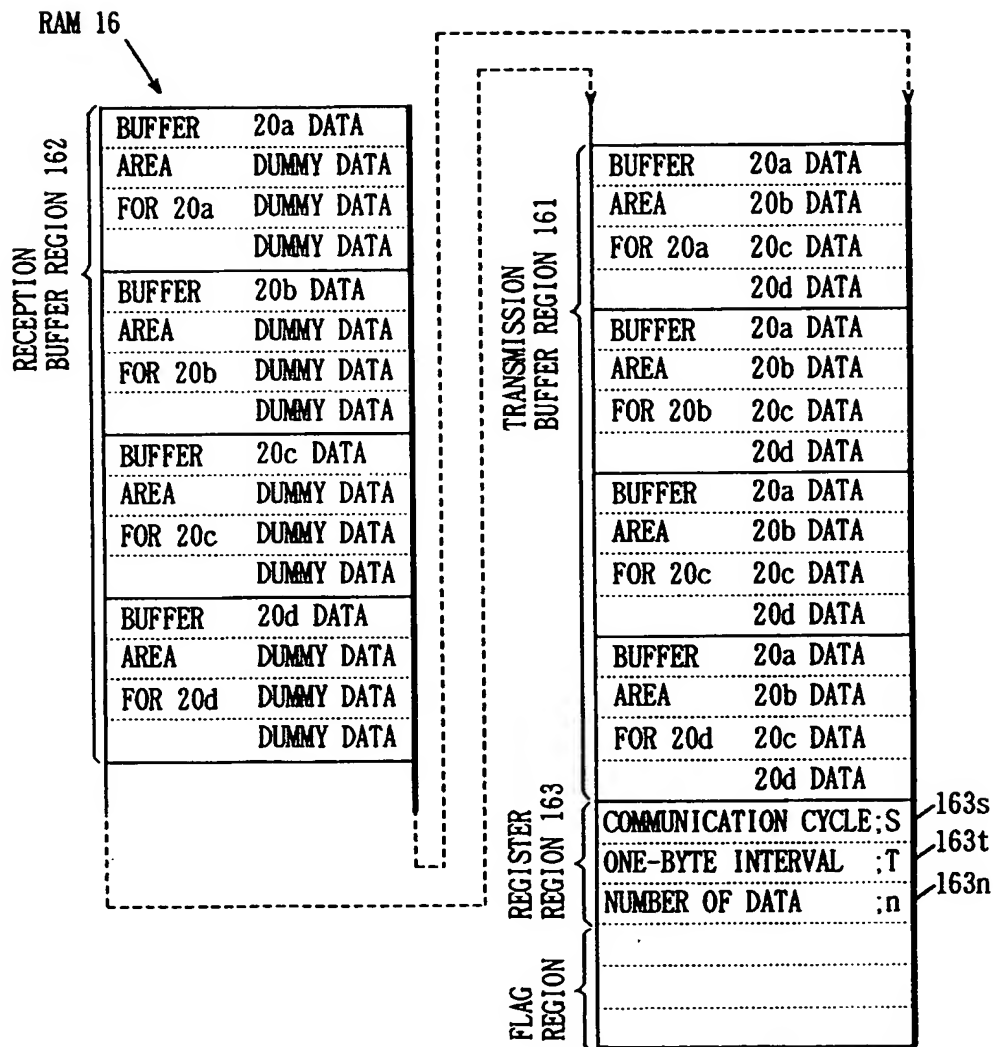




FIG. 5

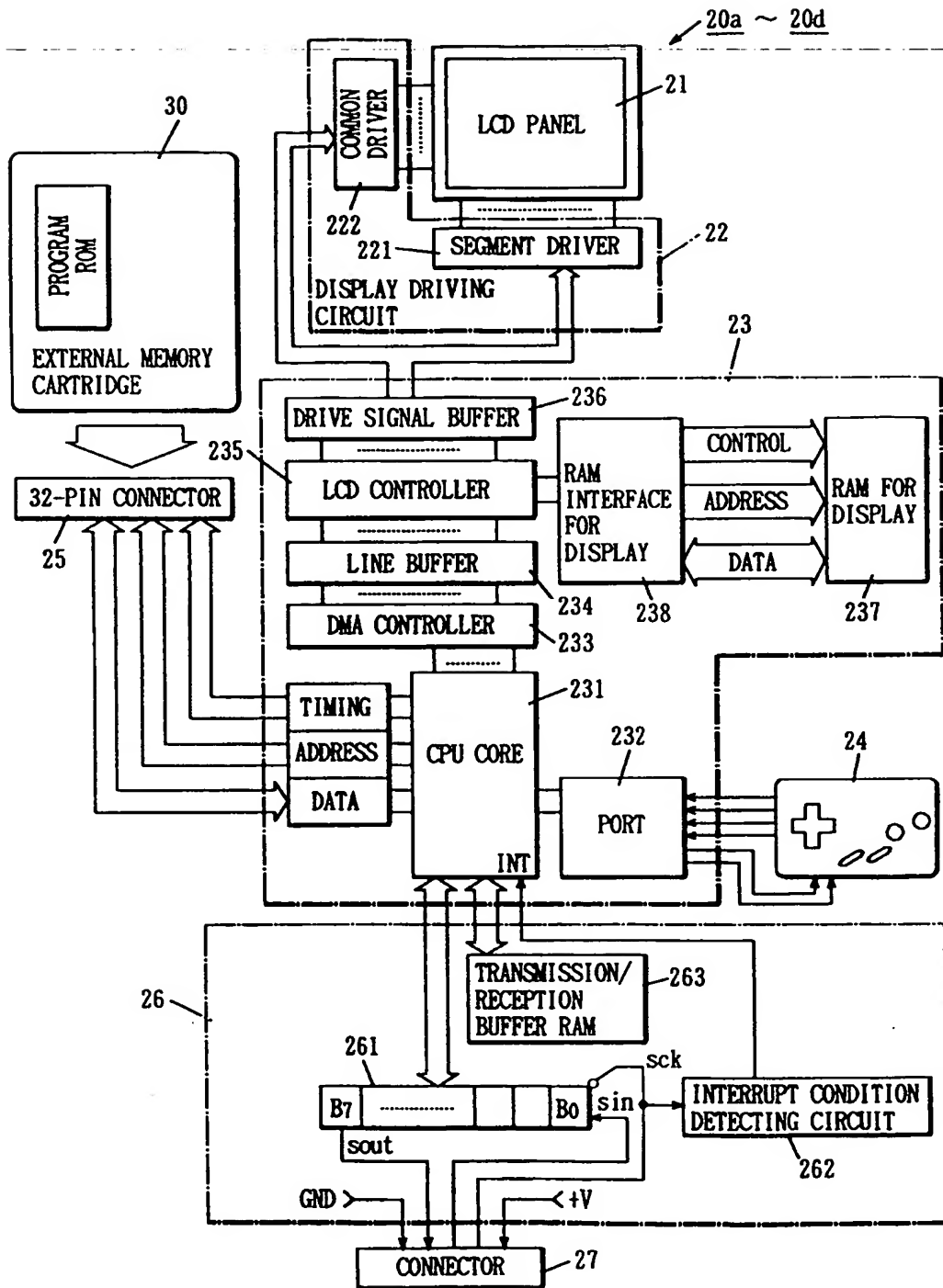


FIG. 6A

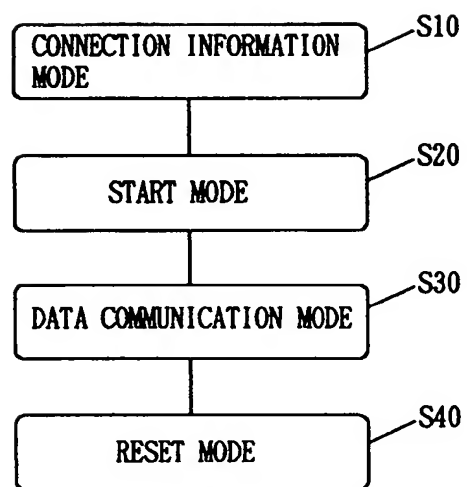


FIG. 6 B

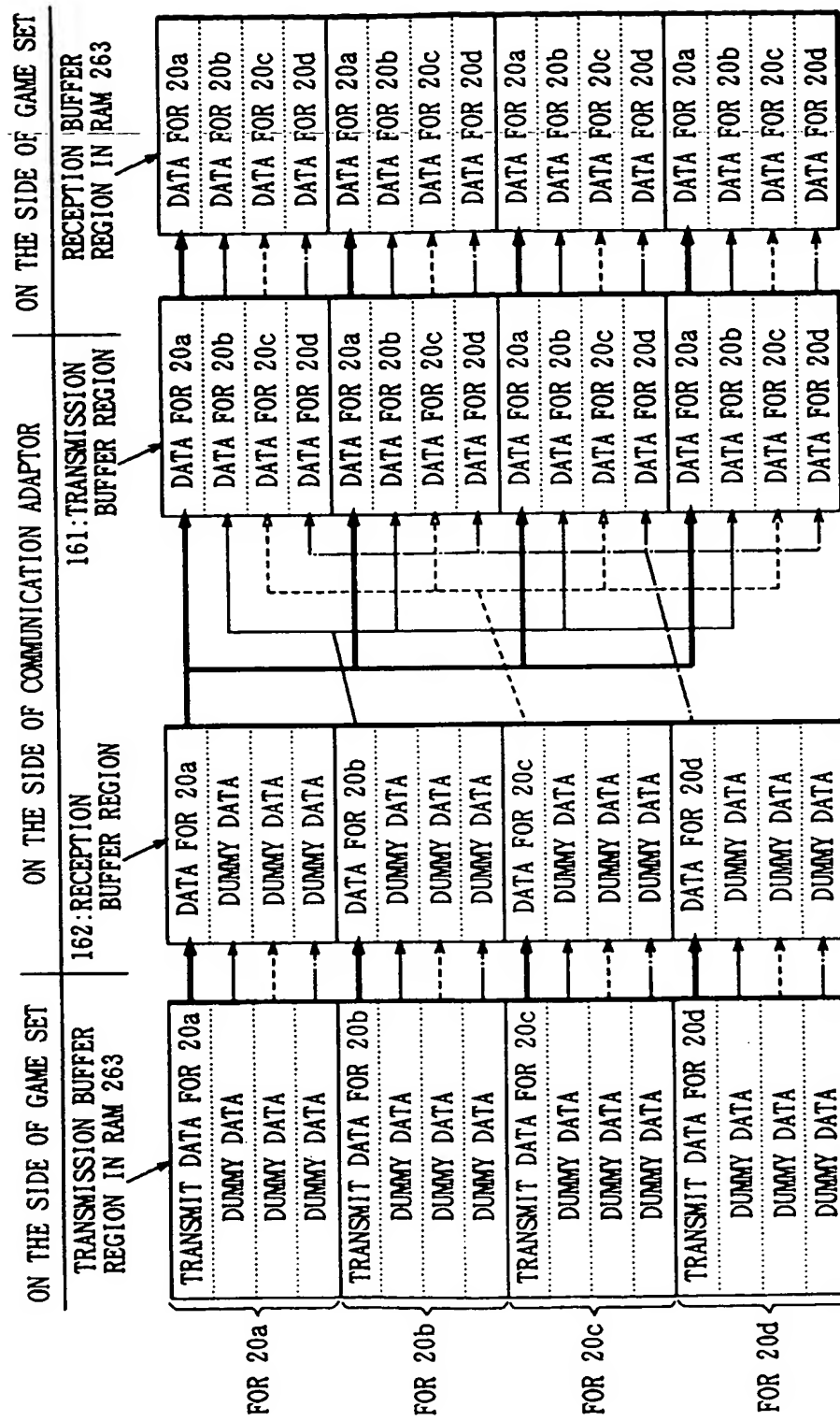


FIG. 7A

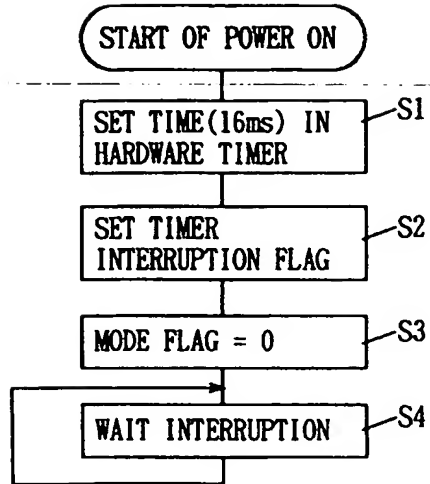


FIG. 7B

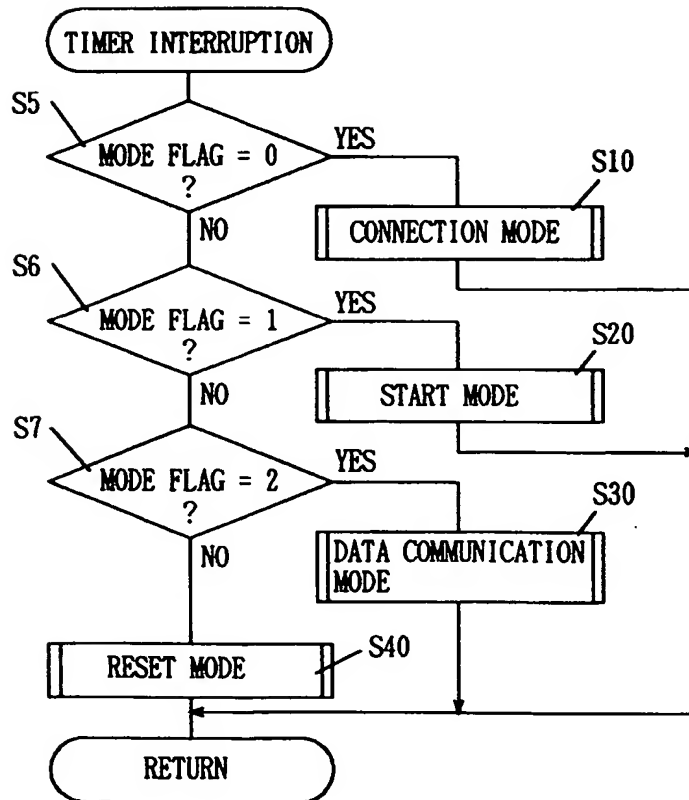


FIG. 7C

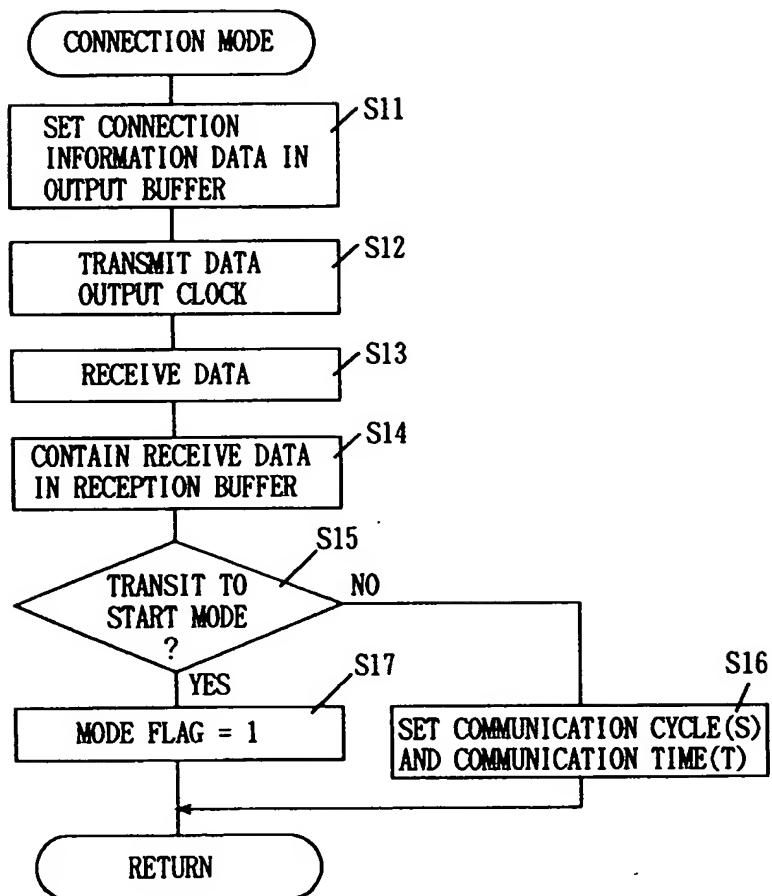


FIG. 7D

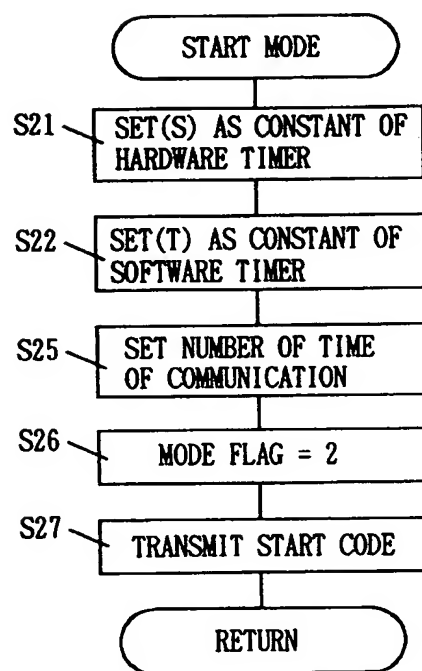


FIG. 7E

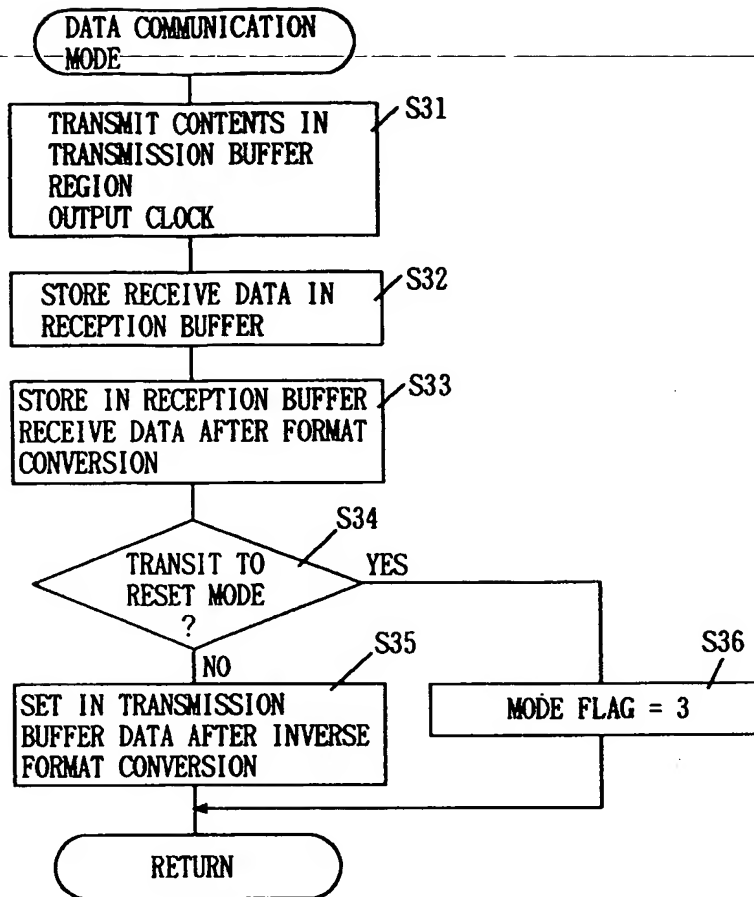


FIG. 7F

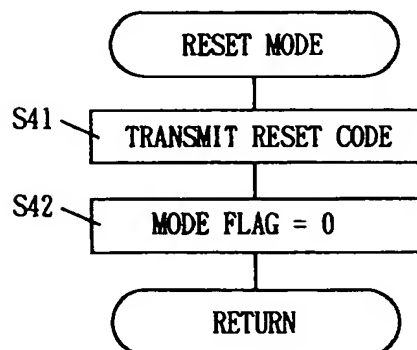




FIG. 8A

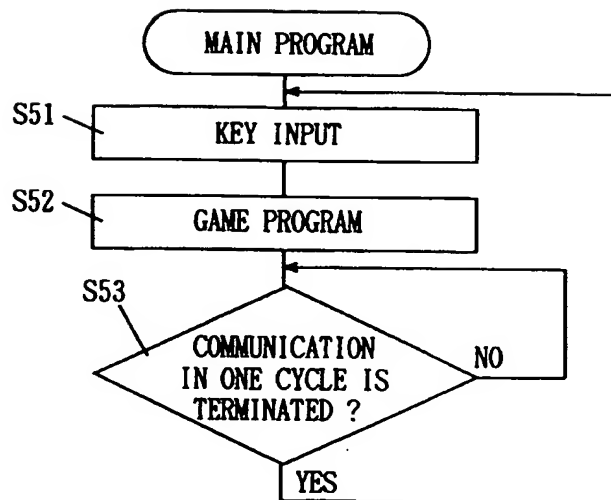


FIG. 8B

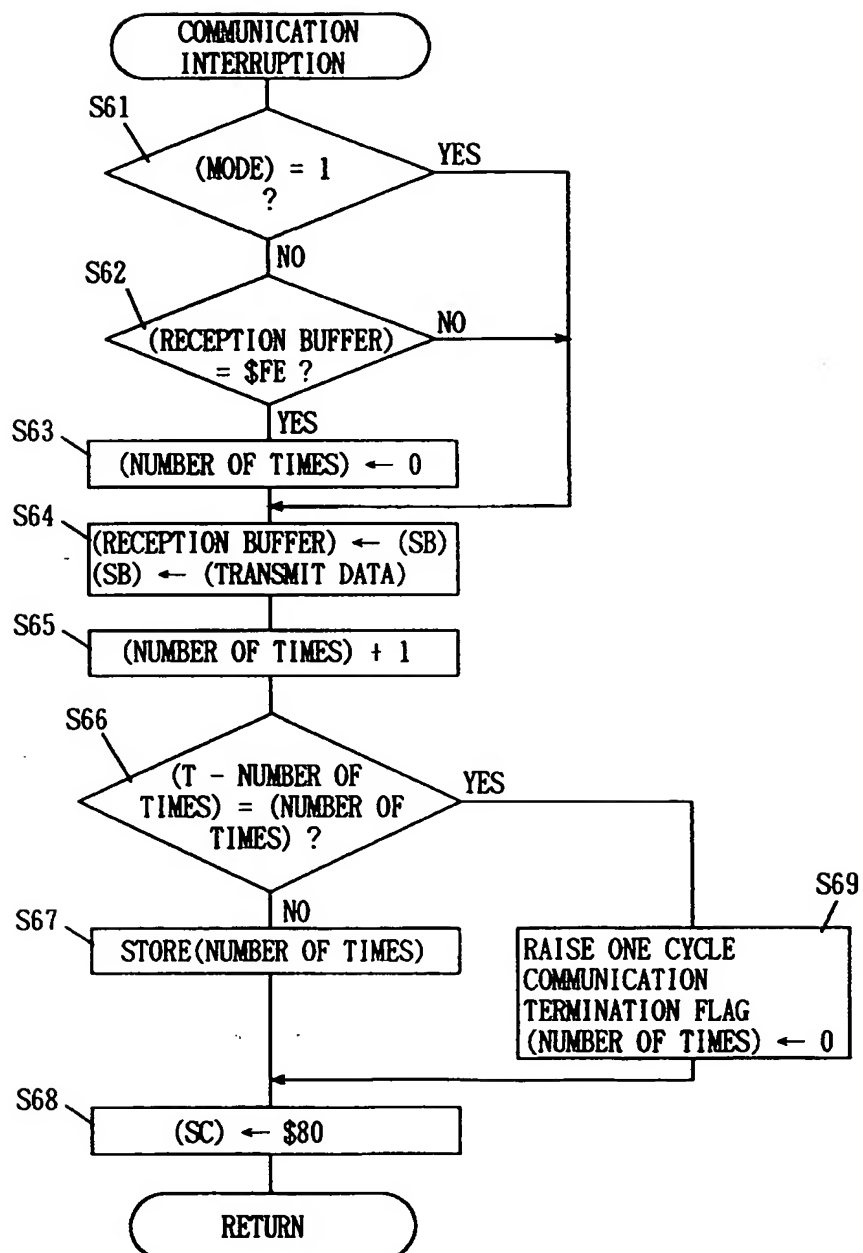


FIG. 9A

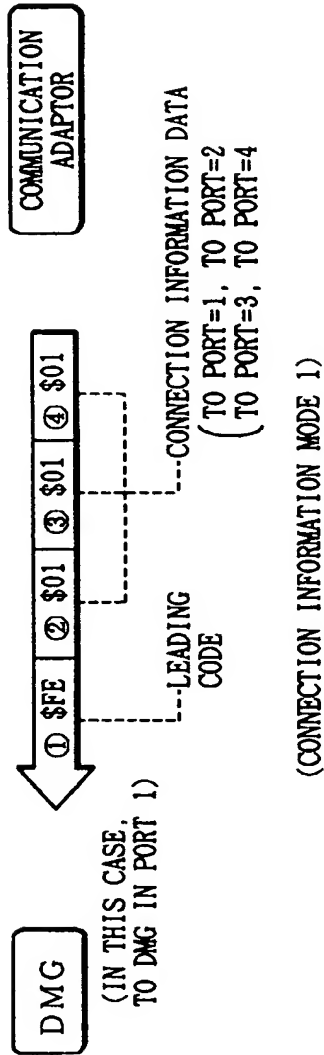


FIG. 9B

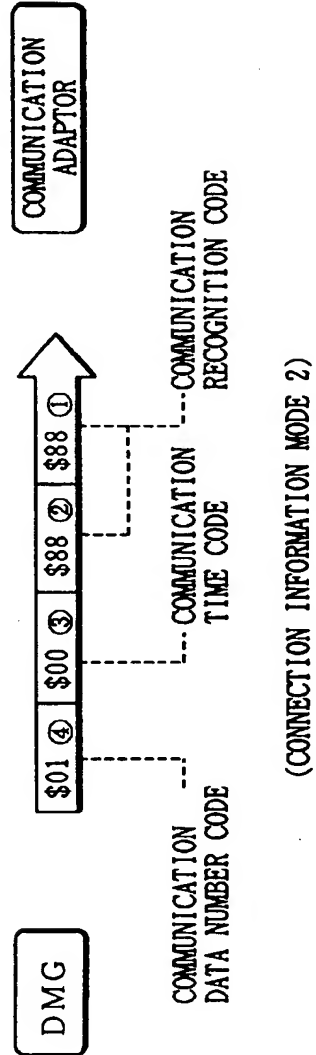


FIG. 9C

CONNECTING STATE					NUMBER OF PORT			
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	1	→ TO DMG IN PORT 1
0	0	0	0	0	0	1	0	→ TO DMG IN PORT 2
0	0	0	0	0	0	1	1	→ TO DMG IN PORT 3
0	0	0	0	0	1	0	0	→ TO DMG IN PORT 4

PRESENCE OR ABSENCE OF  
CONNECTION OF EACH DMG  
0: ABSENCE 1: PRESENCE

(BIT CONFIGURATION OF CONNECTION INFORMATION DATA)

FIG. 10A

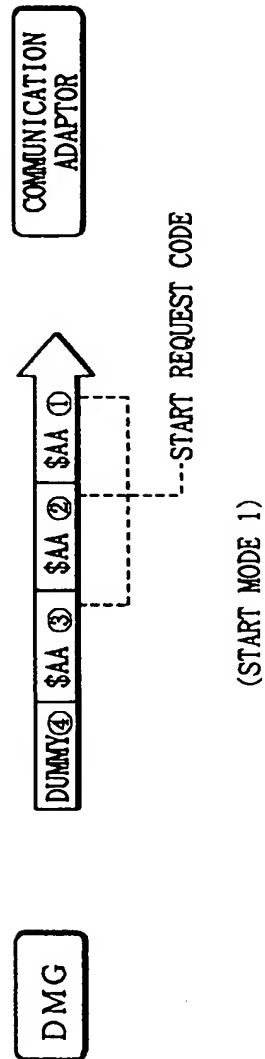


FIG. 10B

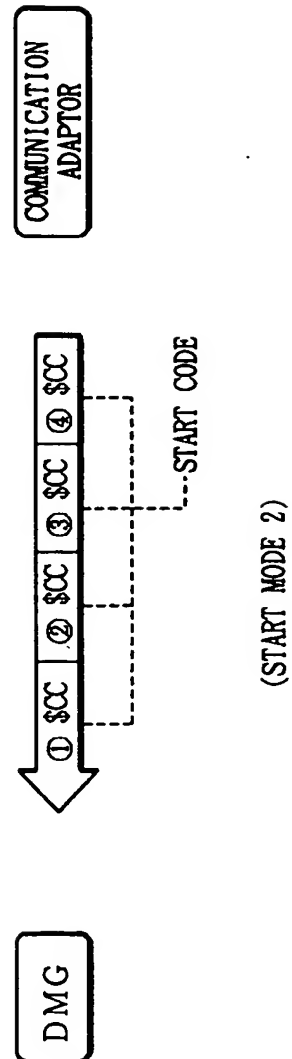
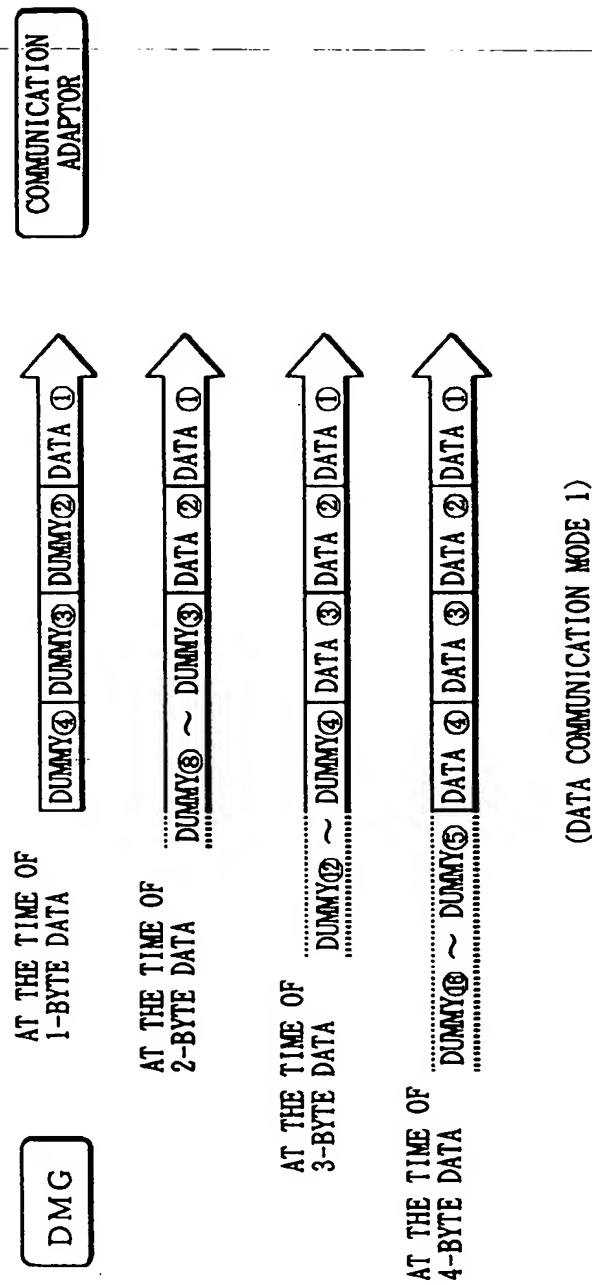


FIG. 11A



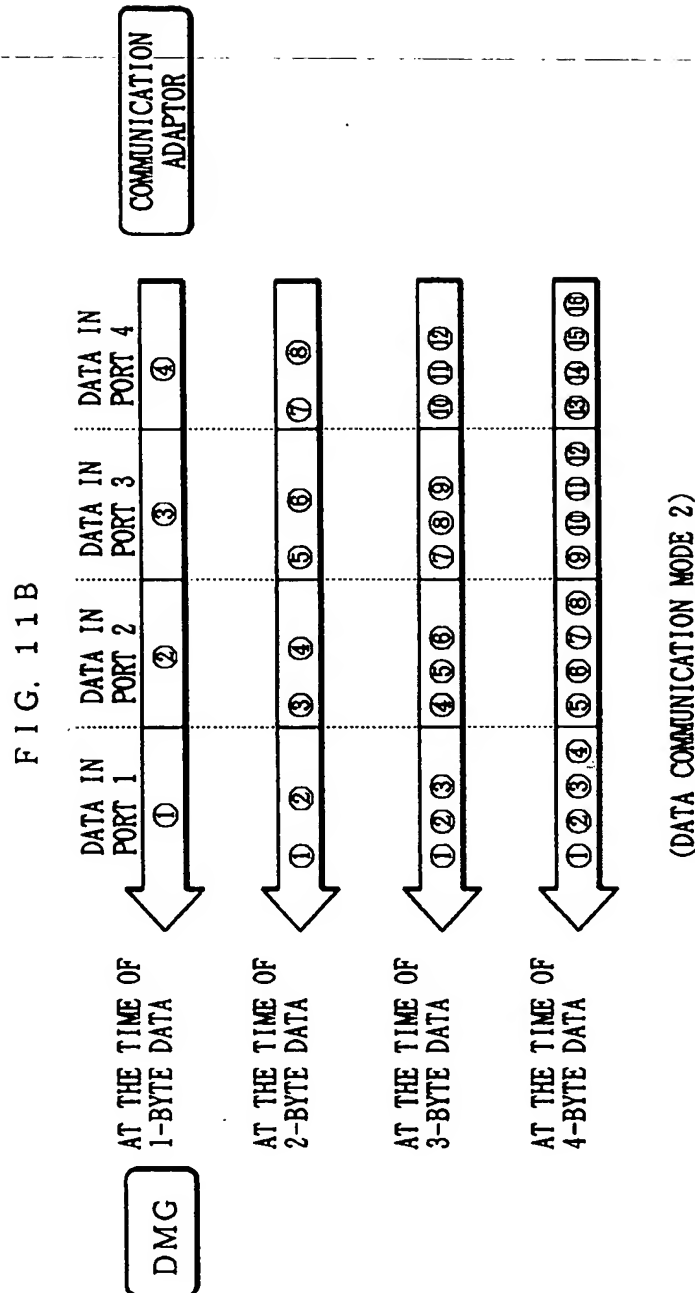




FIG. 12A

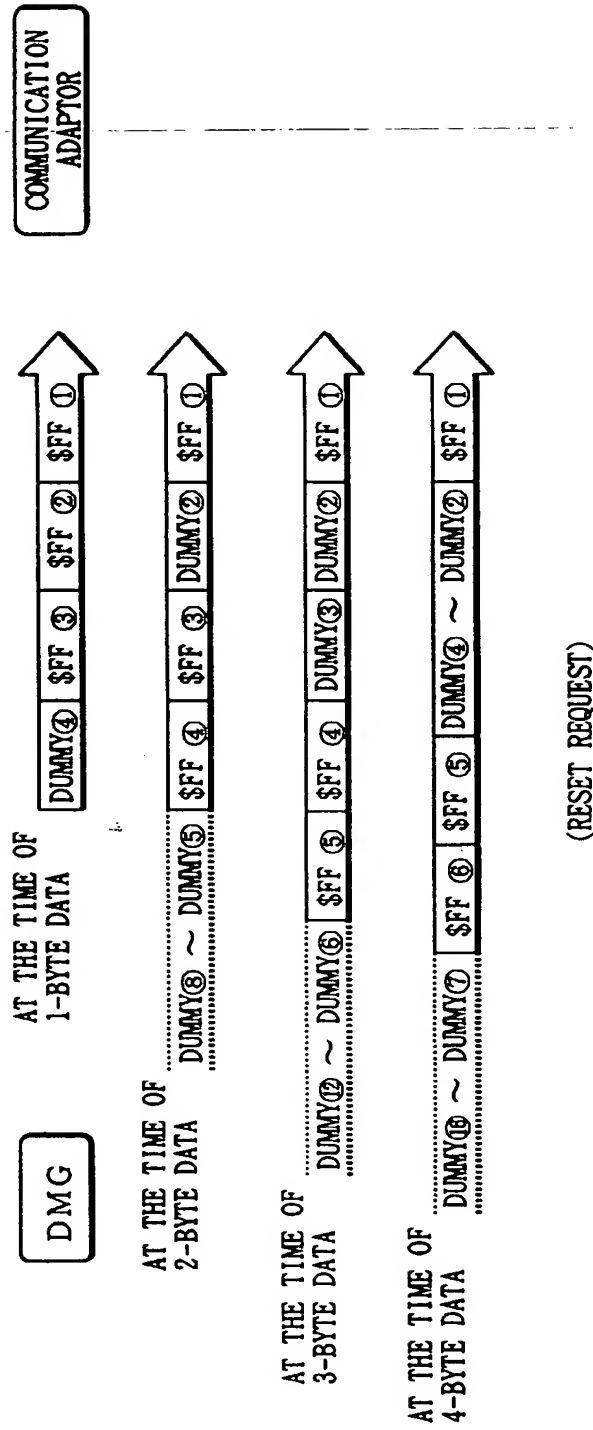


FIG. 12B

